ENTEGRASYON SERVİSİ

IntegService API Kılavuzu

Kimlik doğrulama, yetkiler, uç noktalar, örnek istek ve yanıtlar

DBHSOFT · Sürüm 1.0

18.11.2026

İçindekiler

Genel bakış

1. Kimlik doğrulama (token alma)

2. Yetkiler

3. Fiş export isteği

4. Örnek istekler

5. Yanıt yapısı

6. Hata kodları ve sık karşılaşılan durumlar

7. Faturalar ve siparişler

8. Müşteriler, raporlar ve özellikler

9. Ödemeler

10. Ürünler

11. Stoklar ve token kontrolü

Genel bakış



Token süresi dolunca /api/v1/account/check yeni token döner; ya da yeniden authenticate çağrılır.

IntegService, SGO'daki fiş, müşteri, ödeme, ürün ve stok verilerini dış sistemlere açan REST servisidir. Tüm metodlar aynı token ve yetki kurallarıyla çalışır (bölüm 1 ve 2); 3–6. bölümler fiş export metodunu, 7–11. bölümler diğer metodları anlatır.

Fiş export servisi, SGO'da kesilen faturaları, irsaliyeleri ve iadeleri kalemleri ve iskontolarıyla birlikte JSON olarak döndürür. Kullanım iki adımdır: önce kullanıcı adı ve şifreyle token alınır, sonra bu token ile fişler sorgulanır.

Grup	Adres öneki	Bölüm
Token alma	`{base_url}/api/v1/account/`	1
Fiş export (`getinv` , `getinvxml`)	`{base_url}/api/v1/integ/invoices/`	3–6
Faturalar ve siparişler	`{base_url}/api/v1/integ/invoices/`	7
Müşteriler, raporlar, özellikler	`{base_url}/api/v1/integ/custs/` , `/reports/` , `/props/`	8
Ödemeler	`{base_url}/api/v1/integ/payments/`	9
Ürünler	`{base_url}/api/v1/integ/metas/`	10
Stoklar ve token kontrolü	`{base_url}/api/stocks/` , `/api/v1/account/check`	11

`{base_url}` , size iletilen servis adresidir (ör. test ve canlı ortam için ayrı adresler). Tüm istek ve yanıtlar `Content-Type: application/json` ile gönderilir ve alınır.

Test ve canlı ortam. Servisin test ve canlı için ayrı adresleri ve ayrı kullanıcıları vardır. Hangi veritabanına bağlanılacağı kullanıcıya göre belirlenir: test kullanıcısı test verisini, canlı kullanıcı canlı veriyi görür. Entegrasyonunuzu önce test ortamında doğrulayın.

Ortak yanıt yapısı. Metodların çoğu aynı zarfı döner:

Alan	Açıklama
`success`	İşlem başarılı mı (`true` / `false`)
`message`	Sonuç mesajı; uyarılar bu metnin sonuna eklenir
`info.company` , `info.country`	Tokenın ait olduğu firma ve ülke kodu

Alan	Açıklama
`data`	Asıl veri (nesne ya da liste)
`count`	Dönen kayıt sayısı; bazı metodlarda her zaman `0` gelir, kayıt sayısı için `data` listesine bakın

Değeri boş (`null`) olan alanlar yanıtta hiç yer almaz. Yanıttaki tarihler `yyyy-MM-dd HH:mm:ss` biçimindedir. İsteklerde tarihleri `yyyy-MM-dd` ya da `yyyy-MM-ddTHH:mm:ss` olarak gönderin; yalnız alan açıklamasında farklı biçim belirtilen metodlar (`gg-aa-yyyy`) istisnadır.

1. Kimlik doğrulama (token alma)

Fiş sorgulamadan önce token alınır; token 30 gün geçerlidir ve her istekte `Authorization` başlığında gönderilir.

POST `{base_url}/api/v1/account/authenticate`

Alan	Tip	Zorunlu	Açıklama
`username`	metin	Evet	Size verilen servis kullanıcı adı
`password`	metin	Evet	Kullanıcı şifresi
`company`	metin	Evet	Firma kodu (ör. `DBH-TEST`)
`country`	metin	Evet	Ülke kodu (ör. `TR`)

Örnek istek:

```
{
  "username": "servis_kullanici",
  "password": "********",
  "company": "DBH-TEST",
  "country": "TR"
}
```

Örnek yanıt:

```
{
  "success": true,
  "message": "New Token",
  "info": { "company": "DBH-TEST", "country": "TR" },
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 ... ",
    "expired": "2026-12-18 13:45:00",
    "expires_in": "2591999.5"
  },
  "count": 0
}
```

Yanıt alanı	Açıklama
`data.token`	Sonraki isteklerde kullanılacak token
`data.expired`	Tokenın geçerliliğinin bittiği zaman (UTC)
`data.expires_in`	Kalan süre, saniye

Token sonraki her istekte şu başlıkla gönderilir (Bearer öneki isteğe bağlıdır): `Authorization: Bearer <token>`. Tokenı tırnak işaretleri olmadan kopyalayın; başında ya da sonunda `"` kalırsa istek 401 ile reddedilir.

2. Yetkiler

Servis kullanıcısının görebileceği veri dört ayrı yetkiyle sınırlanır; yetkiler hesabınız tanımlanırken sizin için ayarlanır. Bir yetki tanımlanmamışsa o alanda kısıt uygulanmaz.

Yetki	Neyi sınırlar	Etkilediği metodlar	Yetki yoksa ne olur
Rol (metod) yetkisi	Hangi metodu çağırabileceğinizi	Tüm metodlar; fiş export için `Integration.Invoices.GetInv`	`403` — `Your role is not authorized.`
Distribütör yetkisi	Hangi distribütörlerin verisini görebileceğinizi	`getinv`, `getinvxml`, `getall`, `getOutletListFull`, ödeme sorgulama metodları	Yetkisiz kodlar sonuçtan çıkarılır (aşağıda)
Fiş tipi yetkisi	Hangi fiş tiplerini okuyabileceğinizi	`getall`	`403` — `Your slip type operation is not authorized for: <tip>.`
Bakiye yetkisi	Müşteri bakiye alanlarını	`getOutletListFull`	Bakiye alanları yanıtta hiç yer almaz

Distribütör yetkisi nasıl uygulanır:

- `dist_codes` göndermezseniz yalnızca yetkili olduğunuz distribütörlerin fişleri döner.
- Gönderdiğiniz kodlardan yetkisiz ya da pasif olanlar sorgudan çıkarılır ve yanıt mesajına uyarı eklenir: `Warning Inactive and unauthorized distributor codes = 001,002`.
- Gönderdiğiniz kodların hiçbiri yetkili değilse yanıt `204 No Content` olur.

Bakiye yetkisi: Yetkiniz yoksa `getOutletListFull` yanıtında `OutletBalance`, `OutletOpenBalance`, `OutletEPoint`, `OutletCheckBalance` ve `OutletBondBalance` alanları bulunmaz. Diğer müşteri alanları değişmez.

Yetki tanımlarında değişiklik istiyorsanız servis sağlayıcınızla iletişime geçin; değişiklikler en geç 1 saat içinde geçerli olur.

3. Fiş export isteği

`getinv`, verilen tarih aralığındaki ve filtrelere uyan fişleri döndürür. Beş alan zorunludur; biri eksikse istek `400` ile reddedilir.

POST `{base_url}/api/v1/integ/invoices/getinv`

Başlıklar: `Authorization: Bearer <token>` ve `Content-Type: application/json`

Alan	Tip	Zorunlu	Açıklama
`doc_type`	sayı	Evet	Fiş tipi, 1–10 (aşağıdaki tablo)
`start_date`	tarih	Evet	Başlangıç tarihi, bu tarih dahil. Eski gg-aa-yyyy biçimi kabul edilmez. Biçim: `2026-10-01` ya da `2026-10-01T00:00:00`
`end_date`	tarih	Evet	Bitiş tarihi, bu tarih **hariç**. 1–31 Ekim için `2026-11-01` gönderin
`exported`	sayı listesi	Evet	Aktarım durumu: `0` = henüz alınmamış, `1` = daha önce alınmış. İkisi için `[0,1]`
`doc_status`	sayı listesi	Evet	Fiş durumu: `0` = İptal, `1` = Onaylı (basılmış), `2` = Onaysız, `3` = Beklemede
`dist_codes`	metin listesi	Hayır	Distribütör kodları. Boşsa yetkili olduğunuz tüm distribütörler
`outlet_code`	metin	Hayır	Müşteri kodu
`outlet_integ_code`	metin	Hayır	Müşterinin entegrasyon kodu 1
`outlet_integ_code_2`	metin	Hayır	Müşterinin entegrasyon kodu 2
`vkn`	metin	Hayır	Müşterinin vergi numarası
`doc_no`	metin	Hayır	Fiş numarası (ör. `TST0001`)
`e_inv`	sayı listesi	Hayır	E-belge türü: 0 = e-arşiv, 1 = e-fatura. Boş bırakılırsa uygulanmaz
`ettn`	metin	Hayır	E-belgenin ETTN numarası
`dell_no`	metin	Hayır	Sevkiyat numarası
`limit`	sayı	Hayır	En fazla kaç fiş döneceği. `0` ya da boş = sunucu üst sınırı (varsayılan 5000)
`update_exported`	doğru/yanlış	Hayır	Varsayılan `true`: dönen fişler "aktarıldı" olarak işaretlenir. Aynı fişleri tekrar almak için `false` gönderin

Fiş tipleri (`doc_type`):

Değer	Fiş tipi	İngilizce adı
1	Satış faturası	Sales Invoice
2	Alış faturası	Buy Invoice
3	Müşteriden sağlam iade faturası	Return Invoice From Outlet

Değer	Fiş tipi	İngilizce adı
4	Müşteriden bozuk iade faturası	Corrupt Return Invoice From Outlet
5	Merkeze sağlam iade faturası	Return Invoice To Headquarter
6	Merkeze bozuk iade faturası	Corrupt Return Invoice To Headquarter
7	Satış irsaliyesi	Sales Waybill
8	Taşıma irsaliyesi	Shipment Waybill
9	Sağlam depo transferi	Stock Transfer
10	Bozuk depo transferi	Corrupt Stock Transfer

> **Dikkat**
> `update_exported` varsayılan olarak açıktır. `exported: [0]` ile yeni fişleri çektiğinizde bu fişler işaretlenir ve bir sonraki `[0]` sorgusunda gelmez. Yalnızca bakmak istiyorsanız `update_exported: false` gönderin.

XML çıktı (`getinvxml`). Aynı gövdeyle `{base_url}/api/v1/integ/invoices/getinvxml` adresi çağrılırsa fişler XML olarak döner. XML metni yanıtın `data` alanındadır; alanlar ve kurallar `getinv` ile aynıdır. Muhasebe programına XML ile aktarım yapan sistemler bu adresi kullanabilir.

4. Örnek istekler

Token alma (curl):

```
curl -X POST "{base_url}/api/v1/account/authenticate" \
  -H "Content-Type: application/json" \
  -d '{"username":"servis_kullanici","password":"********","company":"DBH-TEST","country":"TR"}'
```

Henüz alınmamış satış faturaları (Ekim 2026, onaylı):

```
curl -X POST "{base_url}/api/v1/integ/invoices/getinv" \
  -H "Authorization: Bearer <token>" \
  -H "Content-Type: application/json" \
  -d '{
    "doc_type": 1,
    "start_date": "2026-10-01",
    "end_date": "2026-11-01",
    "exported": [0],
    "doc_status": [1],
    "limit": 100
  }'
```

Tek bir fişi, işaretlemeden tekrar alma:

```
{
  "doc_type": 1,
  "start_date": "2026-10-19",
  "end_date": "2026-10-20",
  "exported": [0, 1],
  "doc_status": [0, 1, 2, 3],
  "dist_codes": ["D001"],
  "doc_no": "TST0001",
  "update_exported": false
}
```

Bir müşterinin iade faturaları (vergi numarasıyla):

```
{
  "doc_type": 3,
  "start_date": "2026-01-01",
  "end_date": "2027-01-01",
  "exported": [0, 1],
  "doc_status": [1],
  "vkn": "1234567890",
  "update_exported": false
}
```

5. Yanıt yapısı

Yanıt, fişleri `data.slips` altında döner; her fişte müşteri bilgisi (`outlets`), kalemler (`items`) ve her kalemin iskontoları (`discounts`) bulunur. Tutarlar TL'dir; aksi belirtilmedikçe KDV hariçtir.

Örnek yanıt (kısaltılmış, TST0001 fişi):

```
{
  "success": true,
  "message": "The transaction is successful.",
  "info": { "company": "DBH-TEST", "country": "TR" },
  "data": {
    "slips": [
      {
        "dist_code": "D001",
        "slip_type": "Normal",
        "slip_id": 100001,
        "slip_number": "TST0001",
        "canceled_slip_value": "Printed",
        "slip_date": "2026-10-19 18:05:00",
        "outlet_name": "DBH TEST MÜŞTERİ",
        "store_code": "91",
        "pay_type_code": 3,
        "gross_amount": 2376.0,
        "total_discount": 1171.28,
        "vat": 12.05,
        "gross_total": 1216.77,
        "due_date": "2026-11-09 00:00:00",
        "seller_code": "175",
        "outlets": { "branch": { "central_code": "C0001", "central_name": "DBH TEST MÜŞTERİ",
"tax_no": "1234567890" } },
        "items": [
          {
            "item_no": 2,
            "product_code": "P0002",
            "product_name": "TEST ÜRÜN B 20'li KL",
            "quant": 40.0,
            "product_unit_price": 17.82,
            "row_price": 712.8,
            "price": 361.416,
            "product_real_unit_price": 9.0354,
            "unit": "Koli",
            "vat": 1.0,
            "discounts": [
              { "item_no": 2, "code": "Cust", "sign_discount": "Customer", "type": "Percent",
"discount_rate": 0.03, "discount_val": 21.384, "total_price": 712.8 },
              { "item_no": 2, "code": "EPoint", "sign_discount": "E Point", "type": "Price",
"discount_rate": 0.4773, "discount_val": 330.0, "total_price": 691.416 }
            ]
          }
        ]
      }
    ],
    "failedSlips": []
  },
  "count": 1
}
```


Genel alanlar

Alan	Açıklama
`success`	İşlem başarılı mı
`message`	Sonuç mesajı; yetkisiz distribütör ya da hatalı fiş varsa uyarı eklenir
`count`	Dönen fiş sayısı

Alan	Açıklama
`data.slips`	Fişler
`data.failedSlips`	Oluşturulamayan fişler: `slip_id`, `slip_number`, `dist_code`, `error`. Bir fişin hatası diğerlerini engellemez

Fiş (`slips[]`)

Alan	Açıklama
`dist_code`	Distribütör kodu
`slip_type`	Fiş türü (`Normal`, `Return` vb.)
`slip_id`	Fişin sistemdeki tekil numarası
`slip_number`	Fiş numarası
`canceled_slip_value`	Fiş durumu: `Canceled` (iptal), `Printed` (onaylı), `Unconfirmed` (onaysız), `Waiting` (beklemede)
`slip_date`	Fiş tarihi ve saati
`slip_code`	Fişin diğer numarası (e-belge numarası vb.)
`outlet_name`, `tax_office`, `tax_code`	Müşteri adı, vergi dairesi, vergi numarası
`store_code`, `store_name`	Çıkan depo
`store_code2`, `store_name2`	Giren depo (transferlerde)
`pay_type_code`	Ödeme tipi kodu
`gross_amount`	Brüt tutar (iskontolardan önce)
`total_discount`	Toplam iskonto tutarı
`vat`	Toplam KDV tutarı
`gross_total`	Fişin ödenecek toplamı (iskontolar düşülmüş, **KDV dahil**)
`due_date`	Vade tarihi
`del_code`, `veh_code`	Dağıtıcı ve araç kodu
`seller_code`, `seller_integ_code`	Satış temsilcisi kodu ve entegrasyon kodu
`slip_explanation`, `slip_explanation2`	Fiş açıklamaları
`outlets.head`, `outlets.branch`	Merkez ve şube müşteri bilgileri (kodlar, adres, kanal, vergi bilgileri)

Kalem (`items[]`)

Alan	Açıklama
`item_no`	Kalem sıra numarası
`product_code`, `product_name`	Ürün kodu ve adı

Alan	Açıklama
`quant`	Miktar, **adet** cinsinden (5 koli × 12 = 60)
`unit`	Fişte kullanılan birim (`Koli` , `Adet`)
`product_unit_price`	Adet başına liste fiyatı
`row_price`	Satır brüt tutarı (`quant` × `product_unit_price`)
`price`	Satır net tutarı (tüm iskontolar düşülmüş, KDV hariç)
`product_real_unit_price`	Adet başına net fiyat (`price` / `quant`)
`vat`	KDV oranı, yüzde (`1.0` = %1)
`minimum_value` , `defined_one_value` , `defined_box_value` , `defined_package_value`	Ürünün birim tanımları
`deposit`	Depozitolu ürünse `1`
`discounts`	Kaleme uygulanan iskontolar, uygulanma sırasıyla

İskonto (`discounts[]`)

Alan	Açıklama
`code` , `sign_discount`	İskonto kodu ve adı (`Cust` = müşteri iskontosu, `EPoint` = e-puan iskontosu)
`type`	`Percent` (yüzde), `Price` (tutar), `Free` (bedelsiz)
`discount_rate`	İskonto oranı, kesir olarak (`0.03` = %3)
`discount_val`	İskonto tutarı
`total_price`	Bu iskontonun uygulandığı tutar (önceki iskontolar düşülmüş)

İskontolar sırayla uygulanır: örnekte %3 müşteri iskontosu 712,80 üzerinden 21,38 TL, e-puan iskontosu kalan 691,42 üzerinden 330,00 TL düşer; satır net tutarı 361,42 TL olur.

Müşteri (`outlets.head` = merkez, `outlets.branch` = şube)

Alan	Açıklama
`integration_code1` , `integration_code2`	Müşterinin entegrasyon kodları
`central_code` , `central_name`	Müşteri (cari) kodu ve adı
`city` , `town` , `adress` , `post_code`	İl, ilçe, adres (alan adı bu yazımla döner), posta kodu
`type_code` , `type_name`	Müşteri tipi
`channel_code` , `channel_name`	Kanal
`national_chain_outlets` , `national_chain_outlets_code` , `national_chain_outlets_name`	Ulusal zincir bilgisi
`telephone` , `fax_number`	İletişim

Alan	Açıklama
`tax_no`, `tax_office`	Vergi numarası ve dairesi
`related_person`	İlgili kişi
`seasonal_outlet`	Dönemsel müşteri
`tax_free`	`1` = vergisiz çalışan müşteri

Kalemin diğer alanları

Alan	Açıklama
`product_type`	Ürün tipi
`weight`, `volume`	Ürün ağırlığı ve hacmi (dm^3)
`minimum_unit`, `minimum_value`	Ürünün ünite ve alt birim değerleri
`defined_one_value`, `defined_box_value`, `defined_package_value`	Adet, kutu ve koli birim adları
`deposit`	`0` = depozitosuz, `1` = depozitolu ürün

Eski TXT/XML aktarım kodları. Muhasebe programına eski TXT/XML formatıyla aktarım yapıyorsanız alanların karşılıkları:

JSON alanı	Eski kod	JSON alanı	Eski kod
`dist_code`	DISTCODE	`item_no`	KALEMNO
`slip_type`	FISTIPI	`product_code`	URUNKODU
`slip_id`	FISID	`product_name`	URUNADI
`slip_number`	FISNO	`quant`	MIKTAR
`canceled_slip_value`	IPTAL	`product_unit_price`	BIRIMFIYAT
`slip_date`	TARIH	`row_price`	SATIRTOPLAMI
`slip_code`	MUSKODU	`price`	FIYAT
`outlet_name`	MUSADI	`unit`	BIRIM
`tax_office`	VERGIDAIRESI	`vat`	KDV
`tax_code`	VERGINO	`product_type`	TIPI
`store_code`, `store_code2`	DEPOKODU, DEPOKODU2	`weight`	AGIRLIK
`pay_type_code`	ODEMETIPI	`volume`	HACIM
`gross_amount`	TOPLAMBRUT	`minimum_unit`	UNITE
`total_discount`	TOPLAMISKONTO	`minimum_value`	ALTBIRIM
`vat`	TOPLAMKDV	`defined_one_value`	BIRIM0
`gross_total`	TOPLAMNET	`defined_box_value`	BIRIM1
`due_date`	VADETARIHI	`defined_package_value`	BIRIM2

JSON alanı	Eski kod	JSON alanı	Eski kod
`del_code`	DAGITICIKODU	`deposit`	DEPOZITO
`veh_code`	ARACKODU	`discounts[].code`	KODU
`seller_code`	SATISTEMSILCISIKODU	`discounts[].sign_discount`	ADI
`slip_explanation`	FISACIKLAMA	`discounts[].discount_rate`	ORAN
`slip_explanation2`	FISACIKLAMA2	`discounts[].total_price`	TUTAR

6. Hata kodları ve sık karşılaşılan durumlar

HTTP kodu	Mesaj	Sebep	Ne yapmalı
`200`	`The transaction is successful.`	İşlem başarılı	—
`200`	`... slip(s) failed, see FailedSlips`	Bazı fişler oluşturulamadı	`data.failedSlips` listesindeki hataları servis sağlayıcınıza iletin
`200`	`Warning Inactive and unauthorized distributor codes = ...`	Gönderilen bazı distribütör kodları yetkisiz ya da pasif	Kodları kontrol edin; diğerlerinin fişleri dönmüştür
`204`	(boş yanıt)	Kriterlere uyan fiş yok ya da gönderilen distribütörlerin hiçbiri yetkili değil	Tarih aralığını, `exported` ve `doc_status` filtrelerini kontrol edin
`400`	`Request değerleri boş`	Zorunlu alanlardan biri eksik (`doc_type` , `start_date` , `end_date` , `exported` , `doc_status`)	Eksik alanı ekleyin
`400`	Alan adıyla birlikte biçim hatası	Bir alan yanlış tipte gönderildi (ör. sayı yerine metin, geçersiz tarih)	Alanı tablodaki tipe göre gönderin
`401`	`Invalid or missing token.`	Token yok, süresi dolmuş ya da hatalı kopyalanmış	Yeniden token alın; başında ve sonunda tırnak olmadığından emin olun
`403`	`Your role is not authorized.`	Kullanıcının bu metodu çağırma yetkisi yok	Servis sağlayıcınızdan `Integration.Invoices.GetInv` yetkisini isteyin

Sık sorulanlar

- **Aynı fiş ikinci sorguda gelmiyor.** `update_exported` varsayılan olarak açık olduğu için fiş ilk sorguda "aktarıldı" olarak işaretlendi. Tekrar almak için `exported: [0, 1]` gönderin.
- `quant` **ile ekrandaki miktar farklı.** `quant` her zaman adet cinsindendir; ekranda koli görünebilir. Koli sayısı = `quant` / koli içi adet.
- **Bitiş tarihindeki fişler gelmiyor.** `end_date` dahil değildir; bir sonraki günü gönderin.

7. Faturalar ve siparişler

Bu metodlar fiş (fatura, sipariş, irsaliye) okur, oluşturur, revize eder ve durumunu değiştirir. Hepsi `POST`'tur, adres öneki `{base_url}/api/v1/integ/invoices`.

Metod	Adres	İşlev	Gereken rol
`getall`	`/getall`	Henüz alınmamış fişleri kalemleriyle döner	`Integration.Invoices.Get`
`bill`	`/bill`	Yeni fiş oluşturur	`Integration.Invoices.bill`
Revize	`` (önek, ek yok)	Mevcut fişleri revize eder (liste)	`Integration.Invoices.Update`
`cancel`	`/cancel`	Fişi numara ve distribütörle iptal eder	`Integration.Invoices.Cancel`
`cancelbyid`	`/cancelbyid`	Fişi ID ile iptal eder	`Integration.Invoices.CancelByIDInv`
`confirmbyid`	`/confirmbyid`	Fişi onaylar	`Integration.Invoices.ConfirmByIDInv`
`unconfirmbyid`	`/unconfirmbyid`	Fişi onaysız duruma alır	`Integration.Invoices.UnconfirmByIDInv`
`getInvStatus`	`/getInvStatus`	Fişlerin durumunu ve tarihlerini döner	`Integration.Invoices.GetInvStatus`

`Integration.Invoices` rolü bu metodların hepsini açar.

Fiş tipleri (`invShapes` ve `targetSlipType`; büyük harfle, birebir yazılır)

Ad	Anlamı	Müşteri ve satıcı kodu
`BUYORDER`	Merkeze verilen alış siparişi	Gerekmez
`BUYORDEROUTLETDETAILED`	Müşteri detaylı alış siparişi	Gerekir
`ORDER`	Müşteri satış siparişi	Gerekir
`VOLUMETRICORDER`	Hacimli sipariş	Gerekir
`NCORDER`	Merkezi tip sipariş	Gerekir
`AVAILABILITY`	Bulunurluk kaydı	Gerekir
`SALESINVOICE`	Satış faturası	Gerekir
`HOTSALESINVOICE`	Sıcak satış faturası	Gerekir
`BUYINVOICEFROMHEADQUARTER`	Merkezden alış faturası	Gerekmez
`BUYINVOICE`	Alış faturası	Gerekir
`SERVICEINVOICE`	Hizmet faturası	Gerekir
`RETURNINVOICETOHEADQ`	Merkeze sağlam iade faturası	Gerekmez

Ad	Anlamı	Müşteri ve satıcı kodu
`CORRUPTRETURNINVOICETOHEADQ`	Merkeze bozuk iade faturası	Gerekmez
`NCRETURNINVOICETOHEADQ`	Merkezi tip merkeze iade faturası	Gerekir
`RETURNINVOICEFROMOUTLET`	Müşteriden sağlam iade faturası	Gerekir
`CORRUPTRETURNINVOICEFROMOUTLET`	Müşteriden bozuk iade faturası	Gerekir
`SALESWAYBILL`	Satış irsaliyesi	Gerekir
`HOTSALESWAYBILL`	Sıcak satış irsaliyesi	Gerekir
`NCWAYBILL`	Merkezi tip irsaliye	Gerekir
`BUYWAYBILL`	Alış irsaliyesi	Gerekmez
`RETURNWAYBILLFROMOUTLET`	Müşteriden sağlam iade irsaliyesi	Gerekir
`CORRUPTRETURNWAYBILLFROMOUTLET`	Müşteriden bozuk iade irsaliyesi	Gerekir

Fiş durumları (`docStatus`): `Canceled` = iptal, `Printed` = numaralı/onaylı, `Unconfirmed` = onaysız, `Waiting` = onaylı, beklemede.

7.1 getall

Fiş tipi ve distribütör yetkileri uygulanır (bölüm 2). Yetkisi olmayan fiş tipi istenirse `403` döner.

Alan	Tip	Zorunlu	Açıklama
`startDate`	tarih	Evet	Başlangıç, dahil
`endDate`	tarih	Evet	Bitiş, **hariç**
`invShapes`	metin listesi	Hayır	Fiş tipleri; varsayılan `["BUYORDER"]`
`distCode` / `distCodes`	metin / metin listesi	Hayır	Distribütör kodları
`limit`	sayı	Hayır	En fazla fiş; boş = 5000
`exported`	doğru/yanlış	Hayır	Boşsa yalnız alınmamış fişler; `true` = daha önce alınmış fişler
`includeUnconfirmed`	doğru/yanlış	Hayır	`true` = onaysız fişler de gelir
`ignoreSellerLock`	doğru/yanlış	Hayır	`true` = gün sonu kapatılmamış satıcıların mobil siparişleri de gelir

```
{ "startDate": "2026-11-01", "endDate": "2026-11-18", "invShapes": ["BUYORDER", "ORDER"],
"distCodes": ["1001"], "limit": 500 }
```

Yanıtta her fiş için `idInv` , `shapeName` , `docStatus` , `invoiceNumber` , `distCode` , `evDate` , `shipmentDate` , müşteri ve şube bilgileri (`customerCode` , `customerName` , `customerBranchCode`), `sellerCode` , depo bilgileri, tutarlar (`gross` , `discount` , `kdv` , `price`), açıklamalar, `items[]` (`metaCode` , `amount` , `idUnit` , `unitName` , `unitPrice` , `realPrice` , `discount` , `itemNo`) ve `gainedDiscs[]` (iskonto kodu, adı, tutarı ve yüzdesi) bulunur.

> **Dikkat**

> Dönen fişler "alındı" olarak işaretlenir ve bir sonraki çağrıda gelmez. Aynı fişleri tekrar almak için `exported: true` gönderin.

7.2 bill (fiş oluşturma)

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`centralCode`	metin	Evet	Merkez kodu
`invoiceNumber`	metin	Evet	Fiş numarası; sipariş dışı tiplerde distribütör içinde tekil olmalı
`evDate`	tarih	Evet	Fiş tarihi
`items`	liste	Evet	Kalemler (aşağıda)
`targetSlipType`	metin	Hayır	Fiş tipi; varsayılan `BUYINVOICEFROMHEADQUARTER`
`customerCode` ya da `customerIntegCode`	metin	Fiş tipine göre	Tablodaki "Gerekir" tiplerde biri zorunlu
`sellerCode` ya da `sellerIntegCode`	metin	Fiş tipine göre	Müşteri koduyla aynı kural
`fromStockCode`, `toStockCode`	metin	Hayır	Çıkış ve giriş deposu; boşsa varsayılan depo
`invoiceNumberOther`	metin	Hayır	İkinci numara; varsayılan `invoiceNumber`
`docStatus`	metin	Hayır	`Printed`, `Unconfirmed` ya da `Waiting`; boşsa tipe göre belirlenir
`paymentType`	metin	Hayır	Ödeme tipi; varsayılan nakit
`gross`, `discount`, `kdv`, `price`, `weight`, `volume`	sayı	Hayır	Başlık tutarları; gönderildiği gibi yazılır, sunucu yeniden hesaplamaz
`explanation`, `explanation2`	metin	Hayır	Açıklamalar

`idInv` ve `cancelReason` alanlarını yeni fişte boş bırakın; bunlar revize içindir. Ek başlık alanları: custDisRat1 (müşteri iskontosu, %), priceCurr (senet tutarı), rebateDis (tabela iskontosu). Ağırlık gram, hacim dm³ cinsindendir.

Kalemler (`items[]`)

Alan	Tip	Zorunlu	Açıklama
`metaCode`	metin	Evet	Ürün kodu
`amount`	sayı	Evet	Miktar, `idUnit` biriminde; 0'dan büyük olmalı
`idUnit`	sayı	Evet*	Birim: `0` adet, `1` kutu, `2` koli; ürüne tanımlı olmalı
`sapUnit`	metin	Hayır	Birim kısa adı; doluysa `idUnit` yerine kullanılır
`itemNo`	metin	Evet	Kalem numarası; boş metin (`""`) gönderilebilir
`explanation`	metin	Evet	Satır açıklaması; boş metin (`""`) gönderilebilir
`unitPrice`	sayı	Hayır	Birim fiyat; 0 ise ürün liste fiyatı

Alan	Tip	Zorunlu	Açıklama
`realPrice`	sayı	Hayır	Net satır tutarı; boşsa fiyat, miktar ve iskontolardan hesaplanır
`unitKdvRat`, `unitKdvRat2`	sayı	Hayır	KDV oranları, kesir olarak (`0.18`)
`unitDisRat2` – `unitDisRat4`	sayı	Hayır	Ek kullanıcı iskonto oranları
`unitTaxBase`, `tax`	sayı	Hayır	Birim vergi matrahı ve vergi tutarı; boşsa 0
`firstAmount`	sayı	Hayır	İlk girilen miktar; varsayılan `amount`
`batch`	metin	Hayır	Parti; ürün `metaCode-batch` koduyla aranır
`disc[]`	liste	Hayır	Satır iskontoları (aşağıda)

* `sapUnit` gönderilmiyorsa.

Satır iskontoları (`items[].disc[]`)

Alan	Tip	Zorunlu	Açıklama
`discCode`	metin	Evet	İskonto kodu
`discName`	metin	Evet	İskonto adı
`discRate`	sayı	Hayır	İskonto oranı, 0 ile 1 arası (`0.05` = %5)
`discAmount`	sayı	Hayır	İskonto tutarı
`discOrderNo`	sayı	Hayır	Uygulama sırası

```
{ "distCode": "1001", "centralCode": "C1001", "invoiceNumber": "F2026000123", "evDate": "2026-11-18 10:00:00", "targetSlipType": "ORDER", "customerCode": "M001", "sellerCode": "S01", "items": [ { "metaCode": "P100", "amount": 10, "idUnit": 2, "unitPrice": 120, "unitKdvRat": 0.18, "itemNo": "10", "disc": [ { "discCode": "D1", "discName": "Kampanya", "discRate": 0.05, "discAmount": 60, "discOrderNo": 1 } ] } ] }
```

Yanıtta `data.idInv` oluşan fişin numarasıdır.

HTTP	Mesaj	Sebep
`400`	`Target slip type is not found`	Geçersiz `targetSlipType`
`400`	`The <kod> Distributor Code was not found!`	Distribütör yok
`400`	`Customer is not found` / `Seller is not found`	Müşteri ya da satıcı kodu gereken tipte gönderilmedi
`400`	`Duplicate invoice: InvNo = X, Dist Code = Y!`	Fiş numarası zaten kullanılmış
`400`	`<kod> Product Code value/s not found!`	Ürün yok
`400`	`The unit of idUnit <n> of the product with code <kod> was not found!`	Birim ürüne tanımlı değil
`400`	`DiscRate must be between 0-1`	Geçersiz iskonto oranı
`400`	`The DistCode field is required.` vb.	Zorunlu alan eksik

7.3 Revize (mevcut fişi güncelleme)

POST `{base_url}/api/v1/integ/invoices` — gövde bir listedir. Her öğe `bill` ile aynı alanları taşır; ek olarak `idInv` (revize edilecek fiş) zorunludur. Eski fiş kapatılır ve yerine ona bağlı yeni bir fiş oluşturulur; fiş tipi eski fişten alınır. Sipariş dışı tiplerde yeni bir `invoiceNumber` gönderilmelidir.

```
[ { "idInv": 605090001, "distCode": "1001", "centralCode": "C1001", "invoiceNumber": "SIP-778",
"items": [ { "metaCode": "P100", "amount": 8, "idUnit": 2, "unitPrice": 120, "unitKdvRat": 0.18 } ]
} ]
```

Yanıt her öğe için ayrı bir sonuç listesidir; `data.idInv` yeni fişin numarasıdır. Bir öğe hata alırsa yanıt `400` olur ama başarılı öğeler kaydedilmiş kalır. `idInv` eksikse `IDInv value must be sent!`, bulunamazsa `Invoice is not found: <id>` döner.

7.4 İptal ve onay

Metod	Gövde	Sonuç
`cancel`	`distCode` (zorunlu), `invoiceNumber` (zorunlu), `cancelReason` (isteğe bağlı iptal sebebi kodu)	Bu servisle oluşturulmuş fişi iptal eder
`cancelbyid`	`idInv` (zorunlu), `cancelReason` (isteğe bağlı)	Fişi iptal eder
`confirmbyid`	`idInv` (zorunlu)	Fişi onaylar: numarası varsa `Printed`, yoksa `Waiting`
`unconfirmbyid`	`idInv` (zorunlu)	Fişi `Unconfirmed` yapar

```
{ "distCode": "1001", "invoiceNumber": "F2026000123", "cancelReason": "IPTAL01" }
```

```
{ "idInv": 605090001 }
```

Başarılı yanıt mesajı işlem yapılan fişi gösterir: `The transaction is successful·– idInv: 605090001`.

HTTP	Mesaj	Sebep
`400`	`No records found with the receipt number X or distributor code Y.`	`cancel`: fiş bulunamadı
`400`	`The receipt number X of the Y distributor code has been canceled already.`	`cancel`: fiş zaten iptal
`400`	`Slip could not be found!`	ID ile fiş bulunamadı
`400`	`Unauthorized operation!`	`confirmbyid` / `unconfirmbyid`: fiş iptal ya da numaralı durumda

7.5 getInvStatus

Fişlerin durumunu döner; fişleri "alındı" olarak işaretlemez.

Alan	Tip	Zorunlu	Açıklama
`startDate`, `endDate`	tarih	Evet*	Tarih aralığı, bitiş hariç
`invShapes`	metin listesi	Hayır	Fiş tipleri; varsayılan `["BUYORDER"]`
`idInv`	sayı	Hayır	Tek fiş
`limit`	sayı	Hayır	Boş = 5000

* `idInv` gönderilmiyorsa.

```
{ "startDate": "2026-11-01", "endDate": "2026-11-18", "invShapes": ["BUYORDER"] }
```

Yanıtta her fiş için `idInv`, `createDate` (kayıt tarihi), `confirmDate` (son değişiklik tarihi) ve `docStatus` döner.

8. Müşteriler, raporlar ve özellikler

Bu metodlar müşteri kartlarını okur, müşteri ekler ya da günceller, cari hesap raporu ve özellik tanımlarını döner. Hepsi `POST` 'tur.

Metod	Adres	İşlev	Gereken rol
`GetOutletList`	`/api/v1/integ/custs/GetOutletList`	Müşteri listesi (rut ve satıcıyla)	`Integration.Custs.GetOutletList`
`getOutletListFull`	`/api/v1/integ/custs/getOutletListFull`	Müşteri kartının tamamı, özellikler ve rutlar	`Integration.Custs.GetOutletListFull`
`GetSaleOutletWise`	`/api/v1/integ/custs/GetSaleOutletWise`	Müşteri ve gün bazında satış miktarı	`Integration.Custs.GetSaleOutletWise`
`customer`	`/api/v1/integ/custs/customer`	Müşteri ekler ya da günceller	`Integration.Custs.Customer`
`getCustomerAccountCumulativeReport`	`/api/v1/integ/reports/getCustomerAccountCumulativeReport`	Cari hesap kümülatif raporu	`Integration.Reports.GetCustomerAccountCumulativeReport`
`list`	`/api/v1/integ/props/list`	Özellik tanımları ve değerleri	`Integration.Props.GetProps`

8.1 GetOutletList ve getOutletListFull

Alan	Tip	Zorunlu	Açıklama
`distCodes`	metin listesi	Hayır	Distribütör kodları

```
{ "distCodes": ["194"] }
```

- `GetOutletList` her müşteri için kod, ad, adres, GPS, ilgili kişi, distribütör, rut ve satıcı bilgisini döner. Birden fazla rutu olan müşteri her rut için ayrı satırda gelir. Sonuç boşsa `data: []` döner.
- `getOutletListFull` müşteri kartının tamamını döner: kodlar, vergi bilgileri, tip ve kanal, durum (`Active` / `Passive` / `Disabled`), kredi bilgileri (`creditLimit` , `creditCode` , `creditControlStatus`), ödeme tipi, vade, iletişim ve adres, GPS, bakiyeler, `customerProperties[]` (özellik kodu ve değeri) ve `customerRoutes[]` (satıcı, rut, ziyaret günü ve periyodu). Sonuç boşsa `204` döner.
- `getOutletListFull` 'da distribütör ve bakiye yetkileri uygulanır; bakiye yetkisi olmayan kullanıcıda `outletBalance` , `outletOpenBalance` , `outletEPoint` , `outletCheckBalance` , `outletBondBalance` alanları gelmez (bölüm 2). `GetOutletList` 'te distribütör yetkisi uygulanmaz.

8.2 GetSaleOutletWise

Alan	Tip	Zorunlu	Açıklama
`startDate`	tarih	Evet	Başlangıç, dahil
`finishDate`	tarih	Evet	Bitiş, **hariç**
`distCodes`	metin listesi	Hayır	Distribütör kodları

```
{ "startDate": "2026-10-01", "finishDate": "2026-11-01", "distCodes": ["194"] }
```

Yanıtta her satır `distributorId`, `outletId`, `monthNo` (ayın kısa adı, ör. `Oct`), `saleDate` ve `sdCases` (iadeler düşülmüş net satış, koli cinsinden) içerir. Tarih eksikse `422` ve `Parameter (StartDate) cannot be null.` döner.

8.3 customer (müşteri ekleme ve güncelleme)

Müşteri `distributorCode` ve `code` ile aranır. Bulunamazsa eklenir; bulunursa yalnızca gönderilen alanlar güncellenir. Müşteri kodu bu metodla değiştirilemez.

Alan	Tip	Zorunlu	Açıklama
`distributorCode`	metin	Evet	Distribütör kodu
`code`	metin	Evet	Müşteri kodu
`name`	metin	Yeni müşteride	Unvan
`shortName`	metin	Yeni müşteride	Kısa ad
`customerType`	metin	Yeni müşteride	Müşteri tipi adı (sistemde tanımlı adla aynı)
`creditCode`	metin	Yeni müşteride	`Advance` = peşin, `Cash` = nakit, `Deferred` = vadeli
`paymentType`	metin	Yeni müşteride	`Cash`, `Credit Card`, `Bank Transfer`, `Check`, `Bond`, `Open Account`
`creditStatus`	metin	Hayır	`Unlimited`, `Warning`, `Error`
`creditLimit`, `dueDays`	sayı	Hayır	Kredi limiti, vade günü
`mainCustomerCode`	metin	Hayır	Bağlı olduğu merkez müşteri
`integrationCode`, `integrationCode2`	metin	Hayır	Entegrasyon kodları
`sellerDistributorCode`, `sellerCode`	metin	Hayır	Kartın satış temsilcisi
`taxOffice`, `taxNo`, `idNo`, `taxTypeName`	metin	Hayır	Vergi dairesi, vergi no, TC kimlik no, vergi sınıfı
`countryName`, `cityName`, `townName`	metin	Hayır	Adres, sistemde tanımlı adlarla

Alan	Tip	Zorunlu	Açıklama
`address1`, `address2`, `postCode`	metin	Hayır	Adres
`workTel1`, `mobileTel1`, `email` vb.	metin	Hayır	İletişim
`gpsCoordX`, `gpsCoordY`	sayı	Hayır	Boylam, enlem
`passive`, `dontSell`, `disabled`	doğru/yanlış	Hayır	Pasif, satış yapılmaz, iptal

Değerler büyük/küçük harf ve boşluk dahil birebir yazılmalıdır; tanınmayan değer varsayılana döner (`Cash`, `Advance`, `Unlimited`). Yeni müşteride gönderilmeyen tarih alanları bugünün tarihini alır.

```
{ "distributorCode": "194", "code": "C-1001", "name": "Örnek Gıda Ltd.", "shortName": "Örnek Gıda", "customerType": "Bakkal", "creditCode": "Deferred", "paymentType": "Open Account", "creditLimit": 50000, "dueDays": 30, "cityName": "İstanbul", "townName": "Kadıköy", "sellerDistributorCode": "194", "sellerCode": "001" }
```

Yanıtta gönderilen kayıt geri döner. Yeni müşteride eksik alan için `400` ve `You must send Credit Code`, `You must send Payment Type`, `You must send Customer Type`, `You must send Name for Customer` ya da `You must send Short Name for Customer` döner.

8.4 getCustomerAccountCumulativeReport

Alan adları alt çizgilidir.

Alan	Tip	Zorunlu	Açıklama
`start_date`	metin	Evet	Dönem başı, `gg-aa-yyyy`, dahil
`end_date`	metin	Evet	Dönem sonu, `gg-aa-yyyy`, hariç
`limit`	sayı	Evet	Dönecek hesap sayısı (en fazla 50000); gönderilmezse hesap listesi boş gelir
`dist_codes`	metin listesi	Evet	Distribütör kodları
`consider_chain_stores`	sayı	Evet	`0` = şube bazında, `1` = zincir merkezi altında topla
`outlet_codes`	metin listesi	Hayır	Müşteri kodları
`outlet_shape_codes`, `outlet_shape_names`	metin listesi	Hayır	Müşteri tipi
`outlet_passives`	sayı listesi	Hayır	`1` = pasif müşteriler
`seller_codes`, `seller_integ_codes`	metin listesi	Hayır	Satış temsilcisi
`route_days`	sayı listesi	Hayır	Rut günü, `0` = Pazartesi
`acc_status`	sayı	Hayır	`0` = borçsuz, `1` = borçlu
`max_rec_dept`	doğru/yanlış	Hayır	`true`: son tahsilat ve son fatura tarihleri de hesaplanır

```
{ "dist_codes": ["194"], "start_date": "01-10-2026", "end_date": "01-11-2026", "limit": 1000, "consider_chain_stores": 0, "acc_status": 1, "max_rec_dept": true }
```

Yanıt beş liste içerir: `customers` (müşteri kartı), `customerDists` (distribütörler), `customerAccounts` (`revolve` devreden, `debt` dönem borcu, `receivable` dönem alacağı, `balance` bakiye, `sales` ciro, `credit_limit` , `risk` , `max_receivable_date` , `max_dept_date`), `customerSellers` ve `sellers` .

8.5 props/list

Gövdedeki anahtarlar büyük/küçük harfe duyarlıdır; tam olarak aşağıdaki gibi yazılmalıdır.

Alan	Tip	Zorunlu	Açıklama
`distcode`	metin	Evet	Aktif bir distribütör kodu
`subject`	metin	Evet	Özellik konusu, ör. `Cust` (müşteri), `Meta` (ürün)

```
{ "distcode": "194", "subject": "Cust" }
```

Yanıt, zarf olmadan bir listedir: her özellik için `propCode` , `propName` ve olası değerleri (`items[]`). Anahtar eksikse `400` ve `Discode Value Is Required!` ya da `subject Value Is Required!` döner.

9. Ödemeler

Ödeme metodları tahsilat, ödeme, hizmet faturası ve bakiye eşitleme kayıtlarını okur ve yazar. Tüm metodlar `POST` 'tur, adres öneki `{base_url}/api/v1/integ/payments/` .

Metod	İşlev	Gereken rol
`getPay`	Tahsilat/ödeme fişlerini kalemleriyle döner (eski biçim, alt çizgili alan adları)	`Integration.Payments.GetPayments`
`GetPayment`	Tahsilat/ödeme fişlerini kalemleriyle döner	`Integration.Payments.GetPayment`
`GetServiceInvoice`	Hizmet faturalarını döner	`Integration.Payments.GetServiceInvoice`
`BalanceEqualizing`	Bakiye eşitleme kayıtlarını döner	`Integration.Payments.BalanceEqualizing`
`add`	Tahsilat/ödeme fişi ekler	`Integration.Payments.Add`
`Update`	Tahsilat/ödeme fişini günceller	`Integration.Payments.Update`
`Cancel`	Fişi iptal eder	`Integration.Payments.Cancel`
`AddServiceInvoice`	Hizmet faturası ekler	`Integration.Payments.AddServiceInvoice`
`UpdateServiceInvoice`	Hizmet faturasını günceller	`Integration.Payments.UpdateServiceInvoice`

`Integration.Payments` rolü bu metodların hepsini açar. Okuma metodlarında distribütör yetkisi uygulanır (bölüm 2).

Ortak kodlar

Alan	Değerler
`direction`	`Collection` = tahsilat, `Payment` = ödeme
`paymentType`	`Cash` = nakit, `CreditCard` = kredi kartı, `BankTransfer` = havale, `Check` = çek, `Bond` = senet

Değerler büyük/küçük harfe duyarlıdır; tanınmayan değer `Collection` ve `Cash` kabul edilir.

Okuma metodları kayıtları "aktarıldı" olarak işaretler. `getPay` , `GetPayment` , `GetServiceInvoice` ve `BalanceEqualizing` döndürdükleri kayıtları işaretler. `exported: false` ile tekrar çağırdığınızda bir sonraki sayfa gelir. Güncellenen ya da iptal edilen kayıtlar yeniden "aktarılmadı" durumuna döner ve tekrar gelir.

9.1 getPay

Alan	Tip	Zorunlu	Açıklama
`start_date`	metin	Evet	Başlangıç, biçim `gg-aa-yyyy` (ör. `01-01-2026`), dahil
`end_date`	metin	Evet	Bitiş, biçim `gg-aa-yyyy`, **hariç**
`exported`	sayı listesi	Evet	`0` = alınmamış, `1` = alınmış
`doc_status`	sayı listesi	Evet	`0` = iptal, `1` ve `3` = geçerli
`dist_codes`	metin listesi	Hayır	Distribütör kodları
`outlet_code`, `outlet_integ_code`, `outlet_integ_code_2`, `vkn`	metin	Hayır	Müşteri filtreleri
`doc_no`	metin	Hayır	Bağlı fatura numarası
`receipt_no`	metin	Hayır	Makbuz numarası
`is_service`	sayı	Hayır	`0` = filtre yok, `1` = hizmet, `2` = bakiye eşitleme
`doc_type`	sayı	Hayır	`1` nakit, `2` kredi kartı, `3` havale, `4` çek, `5` senet; verilirse yalnız o tipteki kalemler gelir
`seller_code`, `seller_integ_code`	metin	Hayır	Satış temsilcisi
`limit`	sayı	Hayır	En fazla kayıt; boş = sunucu sınırı (5000)

```
{ "dist_codes": ["131"], "start_date": "01-01-2026", "end_date": "01-02-2026", "exported": [0],
"doc_status": [1, 3], "limit": 100 }
```

Yanıtta her kayıt için `pay_id`, `dist_code`, `outlet_code`, `outlet_name`, `collection_date`, `receipt_number`, `dir` (`COLLECTION` / `PAYMENT`), `shape` (`NORMAL` / `CHECK` / `BONDS`), `price`, `seller_code`, `is_service` ve kalemler (`pay_items[]`: `doc_type`, `pay_type_name`, `amount`, `document_no`, `due_date`, `bank`, `branch`) bulunur.

9.2 GetPayment, GetServiceInvoice, BalanceEqualizing

Üçü aynı istek alanlarını kullanır.

Alan	Tip	Zorunlu	Açıklama
`startDate`	tarih	Evet*	Başlangıç, dahil (`2026-01-01`)
`endDate`	tarih	Evet*	Bitiş, **hariç**
`direction`	metin	Evet (`GetPayment`, `GetServiceInvoice`)	`Collection` / `Payment`. `GetServiceInvoice` her zaman tahsilat döner, yine de gönderilmelidir
`exported`	doğru/yanlış	Hayır	Varsayılan `false`: yalnız alınmamışlar. `true`: yalnız alınmışlar
`distCodes`	metin listesi	Hayır	Distribütör kodları

Alan	Tip	Zorunlu	Açıklama
`idPay`	sayı	Hayır	Tek bir kaydı getirir; verilirse tarih ve `exported` filtreleri kullanılmaz
`limit`	sayı	Hayır	En fazla kayıt; boş = 5000

* `idPay` gönderilmiyorsa zorunludur.

```
{ "distCodes": ["131"], "startDate": "2026-01-01", "endDate": "2026-02-01", "exported": false, "direction": "Collection", "limit": 500 }
```

Metod	Yanıttaki başlıca alanlar
`GetPayment`	`idPay`, `distCode`, `evDate`, `receiptNo`, `sellerCode`, `customerCode`, `dir` (`1` tahsilat, `2` ödeme), `idInv`, `docStatus`, `isCanceled`, `items[]` (`paymentType`, `price`, `documentNo`, `dueDate`, `bank`, `bankOffice`)
`GetServiceInvoice`	`idPay`, `distCode`, `evDate`, `invoiceNumber`, `customerCode`, `price`, `serviceType`
`BalanceEqualizing`	`idPay`, `distCode`, `evDate`, `invoiceNumber`, `customerCode`, `price`, `direction` (`Decrease` / `Increase`), `type` (ör. `Decrease Balance`)

9.3 add ve Update (tahsilat/ödeme yazma)

Alan	Tip	Zorunlu	Açıklama
`idPay`	sayı	Yalnız `Update`	Güncellenecek fiş. `add`'de yok sayılır, yanıtta yeni ID döner
`distCode`	metin	Evet	Distribütör kodu
`customerCode`	metin	Evet	Müşteri kodu
`evDate`	tarih	Evet	Belge tarihi
`items`	liste	Evet	Kalemler; fiş tutarı kalemlerin toplamıdır
`items[].paymentType`	metin	Hayır	Ödeme tipi (Ortak kodlar)
`items[].price`	sayı	Hayır	Kalem tutarı
`items[].explanation`	metin	Hayır	Kalem açıklaması
`sellerCode`	metin	Hayır	Satış temsilcisi
`receiptNumber`	metin	Hayır	Makbuz numarası
`direction`	metin	Hayır	Varsayılan `Collection`
`explanation`	metin	Hayır	Açıklama
`idInv`	sayı	Hayır	Bağlı fatura ID

```
{ "distCode": "131", "customerCode": "M001", "sellerCode": "S01", "evDate": "2026-01-15", "receiptNumber": "TH-12", "direction": "Collection", "items": [ { "paymentType": "Cash", "price": 1000 }, { "paymentType": "CreditCard", "price": 500 } ] }
```

Yanıtta gönderdiğiniz kayıt `data` içinde geri döner; `add`'de `idPay` yeni fişin numarasıdır. `Update` kalemleri silip yeniden yazar ve iptal edilmiş bir fişi yeniden geçerli yapar.

9.4 Cancel

Alan	Tip	Zorunlu	Açıklama
`idPay`	sayı	Evet	İptal edilecek fiş

```
{ "idPay": 5013 }
```

İptal edilen fiş okuma metodlarında `isCanceled: true` ile tekrar gelir.

9.5 AddServiceInvoice ve UpdateServiceInvoice

Alan	Tip	Zorunlu	Açıklama
`idPay`	sayı	Yalnız `UpdateServiceInvoice`	Güncellenecek kayıt
`distCode`	metin	Evet	Distribütör kodu
`customerCode`	metin	Evet	Müşteri kodu
`evDate`	tarih	Evet	Tarih
`serviceType`	metin	Evet	Hizmet adı; sistemde tanımlı hizmet adıyla birebir aynı olmalı
`price`	sayı	Hayır	Tutar
`invoiceNumber`	metin	Hayır	Fatura numarası; doluysa kayıt onaylı, boşsa beklemede oluşur
`sellerCode`, `explanation`, `idInv`	—	Hayır	Temsilci, açıklama, bağlı fatura

```
{ "distCode": "131", "customerCode": "M001", "evDate": "2026-01-20", "invoiceNumber": "HZ-1",
"price": 250, "serviceType": "Raf Kirası" }
```

Hatalar: Okuma metodları bölüm 6'daki kodları kullanır. Yazma metodları (`add`, `Update`, `Cancel`, `AddServiceInvoice`, `UpdateServiceInvoice`) hata durumunda `400` ile yalnız hata metnini döner; `items` gönderilmezse metin `Object reference not set to an instance of an object.` olur.

10. Ürünler

Ürün metodları ürün kartını, birimlerini, fiyatlarını, özelliklerini ve çevirilerini okur ve yazar. Adres öneki `{base_url}/api/v1/integ/metas`.

Metod	Yöntem ve adres	İşlev	Gereken rol
Ürün listesi	POST `/getmetalist`	Tüm ürünler, birimleriyle	`Integration.Metas.Post`
Tek ürün	GET `/{idMeta}`	Ürün ID'siyle tek ürün	`Integration.Metas.GetById`
Ürün birimleri	POST `/metaunits`	Bir ürünün birimleri	`Integration.Metas.GetMetaUnits`
Ürün kaydet	POST `` (önek, ek yok)	Ürün ekler ya da günceller (liste)	`Integration.Metas.Save`
Ürün sil	DELETE `/{idMeta}`	Ürünü kalıcı olarak siler	`Integration.Metas.Delete`
Pasife al / aktif et	POST `/delist`	Ürünü pasife alır ya da aktif eder	`Integration.Metas.DeList`
Fiyatları oku	GET `/prices/{idDist}/{idMeta}`	Ürünün fiyat kayıtları	`Integration.Metas.GetMetaPrices`
Fiyat yaz	POST `/prices`	Fiyat ekler ya da günceller (liste)	`Integration.Metas.AddMetaPrices`
Özellikleri oku	POST `/getprop`	Ürünün özellikleri	`Integration.Metas.GetMetaProps`
Özellik yaz	POST `/addprop`	Ürüne özellik değeri ekler	`Integration.Metas.AddMetaProps`
Çeviri yaz	POST `/translate`	Ürün adı çevirisi (liste)	`Integration.Metas.Translate`
Önerilen sipariş	POST `/suggestedOrder`	Müşteriye önerilen ürün listesi yükler	`Integration.Metas.SuggestedOrder`

`Integration.Metas` rolü bu metodların hepsini açar. Ürün kodu parti (batch) içeriyorsa `KOD-PARTİ` biçiminde yazılır, ör. `P0001-ABC`. Dil kodları: `en`, `tr`, `ru`, `ar`, `id`, `ro`, `de`, `fr`, `dk`, `vn`, `th`, `my`, `fa`.

10.1 Ürün okuma (getmetalist, tek ürün, metaunits)

`getmetalist` gövdesi:

Alan	Tip	Zorunlu	Açıklama
`isoCode`	metin	Hayır	Ürün adlarının dili; varsayılan `en`. Anahtar tam olarak `isoCode` yazılmalı

```
{ "isoCode": "tr" }
```

Yanıtta her ürün için `idMeta`, `metaCode`, `genericCode`, `batch`, `name`, `shortName`, `kdvRatio`, `weight`, `volume`, `barcode`, `disabled`, `price`, `buyPrice`, `unitCase`, `dontSell` ve `units[]` (`idUnit`, `amount`, `name`, `shortName`) gelir. Pasif ürünler de listelenir. Tek ürün metodu aynı alanları birimler olmadan döner; ürün bulunamazsa `data` gelmez.

`metaunits` gövdesi:

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`metaCode`	metin	Evet	Ürün kodu

```
{ "distCode": "TEST", "metaCode": "P0001" }
```

10.2 Ürün kaydetme

Gövde bir listedir. Ürün, distribütörde `metaCode` (ve varsa `batch`) ile aranır; bulunamazsa eklenir, bulunursa güncellenir. **Güncellemede gönderilmeyen alanlar sıfırlanır**, bu yüzden ürünün tüm alanlarını gönderin.

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`metaCode`	metin	Evet	Ürün kodu
`batch`	metin	Hayır	Parti; doluysa ürün kodu `metaCode-batch` olur
`name`, `shortName`	metin	Hayır	Ürün adı ve kısa adı
`idMetaShape`	sayı	Hayır	`1` normal, `2` promosyonel, `3` emanet, `4` rakip
`idMinUnit`, `idDefUnit`	sayı	Hayır	En küçük ve varsayılan birim: `0` adet, `1` kutu, `2` koli
`kdvRatio`, `kdvRatio2`	sayı	Hayır	KDV oranı, kesir olarak (`0.20` = %20)
`weight`, `volume`, `palette`, `unitCase`, `subUnit`, `fullCase`, `alcoholRate`, `epoint`	sayı	Hayır	Fiziksel ve lojistik bilgiler
`barcode`	metin	Hayır	Barkod
`noStock`, `isFinancial`, `dontSell`, `fractional`, `dayMetaControl`	sayı	Hayır	`1` = evet, `0` = hayır
`dayMetaLife`	sayı	Hayır	Raf ömrü, gün
`disabled`	doğru/yanlış	Hayır	Yalnız güncellemede: `true` = pasif
`units[]`	liste	Hayır	Birimler: `idUnit`, `amount` (en küçük birim cinsinden miktar), `name`, `shortName`
`dic[]`	liste	Hayır	Ad çevirileri: `lang`, `value`

Fiyat alanları bu metodla yazılmaz; fiyat için `/prices` kullanın.

```
[{ "distCode": "TEST", "metaCode": "P0001", "idMetaShape": 1, "idMinUnit": 0, "idDefUnit": 2,
"name": "Ürün A", "shortName": "ÜrnA", "kdvRatio": 0.2, "barcode": "869000000001", "dic": [{ "lang":
"en", "value": "Product A" }], "units": [{ "idUnit": 0, "amount": 1, "name": "Adet", "shortName":
"AD" }, { "idUnit": 2, "amount": 12, "name": "Koli", "shortName": "KL" }] }]
```

Yanıt her ürün için ayrı bir sonuç içerir; bir ürünün hatası diğerlerini engellemez. En az bir ürün hata alırsa yanıt `400` olur ve hatalı ürün `success: false` ile işaretlenir.

10.3 Silme ve pasife alma

- **Silme** (`DELETE /{idMeta}`) ürünü stok ve birim kayıtlarıyla birlikte kalıcı olarak siler; geri alınamaz. Ürünü yalnız satıştan kaldırmak için `delist` kullanın.
- **delist** gövdesi bir listedir:

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`metaCode`	metin	Evet	Ürün kodu (partili üründe `KOD-PARTİ`)
`disabled`	doğru/yanlış	Hayır	`true` = pasife al, `false` = aktif et

```
[{ "distCode": "TEST", "metaCode": "P0001", "disabled": true }]
```

10.4 Fiyatlar

`GET /prices/{idDist}/{idMeta}` ürünün fiyat kayıtlarını döner: `firstDate` , `lastDate` , `price` , `buyPrice` , `normalReturnPrice` , `corruptReturnPrice` , `advicePrice` , `ruleNo` .

`POST /prices` gövdesi bir listedir:

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`metaCode`	metin	Evet	Ürün kodu (partisiz)
`firstDate`	tarih	Evet	Fiyatın geçerli olduğu ilk gün
`buyPrice`	sayı	Evet	Fiyat; hem satış hem alış fiyatı olarak yazılır
`normalReturnPrice` , `corruptReturnPrice` , `advicePrice`	sayı	Hayır	Sağlam iade, bozuk iade, tavsiye fiyatı
`batch`	metin	Hayır	Doluysa fiyat partili ürüne yazılır; parti yoksa ana üründen oluşturulur
`degree`	sayı	Hayır	Fiyat derecesi
`customerFilterShape` , `customerFilterValue`	metin	Birlikte	Fiyatı bir müşteri grubuna sınırlar (ör. `DISTRIBUTOR GROUP` ve grup kodu). İkisi boşsa genel fiyattır

```
[{ "distCode": "DBH-TEST", "metaCode": "P0004", "firstDate": "2026-09-01", "buyPrice": 100,
"normalReturnPrice": 100, "corruptReturnPrice": 90, "advicePrice": 120, "degree": 1 }]
```

Bitiş tarihini sunucu hesaplar: yeni fiyat başlayınca aynı ürünün önceki fiyatı kapanır, son fiyat süresiz kalır. Bir listede aynı ürün için birden fazla dönem gönderiyorsanız sıralama `firstDate`'e göre yapılır. Bir öğe hata alırsa işlem durur; ondan önceki öğeler kaydedilmiş kalır.

10.5 Özellikler ve çeviriler

`getprop` gövdesi `{ "distCode", "metaCode" }`'dur. `addprop` gövdesi:

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`metaCode`	metin	Evet	Ürün kodu
`props[].propCode`	metin	Evet	Sistemde tanımlı ürün özelliği kodu
`props[].valueCode`	metin	Evet	Değer kodu, en fazla 50 karakter
`props[].name`	metin	Hayır	Değerin adı

```
{ "distCode": "TEST", "metaCode": "P0003", "props": [{ "propCode": "Category", "valueCode": "41",
"name": "Atıştırmalık" }] }
```

Değer, ürünün partili varyantlarına da yazılır. `translate` gövdesi bir listedir: `distCode`, `metaCode`, `lang` ve `value` (çeviri metni) zorunludur.

```
[{ "distCode": "TEST", "metaCode": "P0001", "lang": "en", "value": "Product A" }]
```

10.6 Önerilen sipariş (suggestedOrder)

Gövde bir listedir. Aynı müşteri ve tip için önceki öneri listesi silinir, yenisi yazılır.

Alan	Tip	Zorunlu	Açıklama
`distCode`	metin	Evet	Distribütör kodu
`outletCode`	metin	Evet	Müşteri kodu
`typeCode`	metin	Evet	`1` temel, `2` stratejik, `3` fırsat
`weekStartDate`, `weekEndDate`	tarih	Evet	Listenin geçerli olduğu hafta
`metaCodeTargets[].metaCode`	metin	Evet	Ürün kodu, en fazla 10 karakter
`metaCodeTargets[].targetCount`	sayı	Hayır	Önerilen miktar
`metaCodeTargets[].targetUnit`	sayı	Hayır	Birim: `0` adet, `1` kutu, `2` koli

```
[{ "distCode": "D001", "outletCode": "M0002", "typeCode": "1", "weekStartDate": "2026-11-23",
"weekEndDate": "2026-11-29", "metaCodeTargets": [{ "metaCode": "P0005", "targetCount": 5,
"targetUnit": 1 }] }]
```

Yanıt her zaman `200` döner. `message` başarılı ve hatalı satır sayılarını verir; `data` bulunamayan kodları listeler (ör. `"999: meta code bulunamadı!"`).

Belirli müşterilere özel ürün entegrasyon metodları bu kılavuzun kapsamında değildir.

11. Stoklar ve token kontrolü

Stok metodlarının adres öneki farklıdır: `{base_url}/api/stocks/` (`/v1/integ` yoktur).

Metod	Yöntem	İşlev	Gereken rol
`PastStockMonitoringReportByProduct`	POST	Bir distribütörün geçmiş tarihli stoku, ürün ve depo kırılımında	`Integration.Stocks.GetPastStockMonitoringReportByProduct`
`GetWarehouseStock`	POST	Depo bazında ürün stokları (koli/kutu/adet)	`Integration.Stocks.GetWarehouseStock`
`GetDistSales`	POST	Tüm distribütörlerin aylık satış ve iade miktarları	`Integration.Stocks.GetDistSales`
`GetDistStock`	GET	Tüm distribütörlerin anlık stokları	`Integration.Stocks.GetDistStock`
`account/check`	POST	Tokenı doğrular, süresi dolmuşsa yeniler	Rol gerekmez

11.1 PastStockMonitoringReportByProduct

Alan adları alt çizgilidir. Stok, verilen günden **önceki** hareketlerden hesaplanır; o günün hareketleri dahil değildir.

Alan	Tip	Zorunlu	Açıklama
`dist_code`	metin	Evet	Tek distribütör kodu; yetkili ve aktif olmalı
`stock_day`	metin	Evet	`gg-aa-yyyy` , `gg/aa/yyyy` ya da `gg.aa.yyyy`
`meta_disabled`	sayı listesi	Evet	`0` = aktif ürünler, `1` = iptal ürünler; `[]` = hepsi
`limit`	sayı	Hayır	En fazla satır; boş = 5000
`product_owner`	metin	Hayır	`HO` = yalnız merkez ürünleri; varsayılan `D` = filtre yok
`stock_codes` , `meta_codes` , `seller_codes`	metin listesi	Hayır	Depo, ürün ve plasiyer filtreleri

```
{ "dist_code": "228", "stock_day": "03-11-2026", "limit": 1000, "meta_disabled": [0],
"product_owner": "HO", "stock_codes": ["91"] }
```

Yanıtta `data.meta_stocks[]` her ürün için `meta` (`meta_code` , `meta_name`) ve `stocks[]` (`store_code` , `store_name` , `seller_code` , `package` koli, `piece` toplam adet, `liter` , `kilogram` , `sell_price` ve `buy_price` toplam tutar) içerir. Hatalı ya da yetkisiz distribütör, eksik tarih gibi durumlarda `400` döner.

11.2 GetWarehouseStock

Alan	Tip	Zorunlu	Açıklama
`date`	tarih	Hayır	Verilirse o tarihten önceki stok; verilmezse anlık stok
`distCodes`	metin listesi	Hayır	Distribütör kodları
`productCodes`	metin listesi	Hayır	Ürün kodları
`showOnlyAvailableStock`	doğru/yanlış	Hayır	`true` = sıfır olmayan stoklar

```
{ "date": "2026-11-01", "distCodes": ["228"], "productCodes": ["P0006"], "showOnlyAvailableStock": true }
```

Yanıtta her satır `distributorCode`, `warehouseCode`, `warehouseName`, `productCode`, `productName`, `evDate`, `unit3Amount` / `unit3Name` (koli), `unit2Amount` / `unit2Name` (kutu), `unit1Amount` / `unit1Name` (adet), `volume` ve `weight` içerir.

11.3 GetDistSales ve GetDistStock

`GetDistSales` gövdesi:

Alan	Tip	Zorunlu	Açıklama
`startDate`	tarih	Evet	Başlangıç, dahil
`endDate`	tarih	Evet	Bitiş, dahil; saat verilmezse o günün 00:00'ı alınır (`2026-10-31T23:59:59` gönderin)

```
{ "startDate": "2026-10-01", "endDate": "2026-10-31T23:59:59" }
```

`GetDistStock` parametre almaz (`GET`). İki metod da aynı alanları döner: `material` (ürün kodu), `customer` (distribütör kodu), `salesorg`, `distR_CHAN`, `date` (ayın ilk günü), `koli`, `adet` (koliye tamamlanmayan adet) ve `kg`. Satışlar artı, iadeler eksi olarak toplanır. Bu iki metod tüm distribütörleri döner; distribütör yetkisi uygulanmaz.

11.4 account/check

POST `{base_url}/api/v1/account/check` — gövde yoktur, token `Authorization` başlığında gönderilir.

Durum	Yanıt
Token geçerli	`success: true`, `message: "Valid Token"`, `data` içinde aynı token ve bitiş zamanı
Tokenın süresi dolmuş	`message: "New Token"` ve `data` içinde **yeni token**. `success` bu durumda `false` döner; yeni tokenı kullanın
Token hatalı	`400`, `message: "Invalid token."` — yeniden `authenticate` çağırın

Servisteki çeviri ve log metodları (`/translate`, `/logs`, `/api/v2/log`) iç sistemler içindir ve bu kılavuzun kapsamında değildir. `stockposition` ve `stockmovement` metodları ise yalnız belirli bir müşteri entegrasyonuna özeldir; ihtiyacınız varsa servis sağlayıcınızdan ayrı doküman isteyin.